Journal of Engineering Faculty, 2(2): 214-222, 2024

Journal of Engineering Faculty

| cumfad.cumhuriyet.edu.tr | Founded: 2023 Available online, ISSN Publisher: Sivas Cumhuriyet Üniversitesi

Multi-Class Classification of ISO/IEC 25010 Software Quality Metrics Using User Feedback

Sedat SÖNMEZ[1,a,*], Hakan KEKÜL[2,b]

[1] *Department of Artificial Intelligence and Data Science, Institute of Science and Technology, Sivas Cumhuriyet University, Sivas, Turkey*
[2] *Department of Software Engineering, Faculty of Technology, Sivas Cumhuriyet University, Sivas, Turkey*
**Corresponding author*

Research Article

History

Received: 20/11/2024
Accepted:10/10/2024

ABSTRACT

Software quality indicates how effective and efficient a software is. Various standards need to be used to evaluate software quality. One of the most important internationally accepted standards for measuring software quality is the ISO/IEC 25010 software quality standard. With this standard, the quality of a software product is evaluated with eight different metrics. These are functional suitability, performance, compatibility, usability, reliability, security, maintainability and portability metrics. In this study, we tried to determine the relationship between user feedback and the metrics of the ISO/IEC 25010 software quality standard. Machine learning (ML) and natural language processing (NLP) techniques were used to classify user comments. After the data preprocessing phase, vectors of user comments were extracted with the Tf-Idf method for NLP. As a machine learning method, classification was made using five different models: Extra Trees Classifier (ETC), Gaussian Process Classifier (GPC), MLP Classifier (MLPC), Bernoulli Naive Bayes Classifier (BNBC) and Support Vector Classifier (SVC). Our goal is to show how quality metrics can be classified into multiple classes using user notifications. The data set used has an unbalanced structure, containing 1681 user comments classified by software experts. Synthetic Minority Oversampling Technique (SMOTE) was used to address the imbalance in the dataset. The results were compared by applying the same classification models to unbalanced and balanced data sets. According to the results obtained, the best classification model is the Extra Trees Classifier model, which provides the highest accuracy rate of 87% according to the SMOTE applied data set. The results show that ML and NLP methods can be used effectively in the classification process of software quality metrics.

Keywords: Machine learning, Natural language processing, ISO/IEC 25010, Smote, Multiclass classification

ISO/IEC 25010 Yazılım Kalite Metriklerinin Kullanıcı Geri Bildirimlerini Kullanarak Çok Sınıflı Sınıflandırılması

ÖZ

Yazılım kalitesi, bir yazılımın ne kadar etkili ve verimli olduğunu gösterir. Yazılım kalitesini değerlendirmek için çeşitli standartların kullanılması gerekmektedir. Yazılım kalitesini ölçmek için uluslararası alanda kabul görmüş en önemli standartlardan biri, ISO/IEC 25010 yazılım kalite standardıdır. Bu standart ile bir yazılım ürününün kalitesi sekiz farklı metrik ile değerlendirilmektedir. Bunlar fonksiyonel uygunluk, performans, uyumluluk, kullanılabilirlik, güvenilirlik, güvenlik, bakım kolaylığı ve taşınabilirlik metrikleridir. Bu çalışmada kullanıcı geri bildirimleri ile, ISO/IEC 25010 yazılım kalite standardının metrikleri arasındaki ilişki tespit edilmeye çalışılmıştır. Kullanıcı yorumlarının sınıflandırılmasında Makine öğrenmesi (ML) ve doğal dil işleme (NLP) teknikleri kullanılmıştır. Veri önişleme aşamasından sonra NLP için Tf-Idf yöntemi ile kullanıcı yorumlarının vektörleri çıkarılmıştır. Makine öğrenmesi yöntemi olarak Ekstra Ağaçlar Sınıflandırıcısı (ETC), Gauss Süreç Sınıflandırıcısı (GPC), MLP Sınıflandırıcısı (MLPC), Bernoulli Naive Bayes Sınıflandırıcısı (BNBC) ve Destek Vektör Sınıflandırıcısı (SVC) olmak üzere beş farklı model kullanılarak sınıflandırma yapılmıştır. Amacımız, kalite metriklerinin kullanıcı bildirimlerini kullanarak çok sınıflı nasıl sınıflandırılabileceğini göstermektir. Kullanılan veri seti yazılım uzmanları tarafından sınıflandırılmış 1681 kullanıcı yorumu içeren dengesiz bir yapıya sahiptir. Veri setindeki dengesizliği gidermek için Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE) kullanılmıştır. Dengesiz ve dengeli veri setlerine aynı sınıflandırma modelleri uygulanarak sonuçlar karşılaştırılmıştır. Elde edilen sonuçlara göre, en iyi sınıflandırma modeli SMOTE uygulanan veri setine göre %87 ile en yüksek doğruluk oranını sağlayan Ekstra Ağaçlar Sınıflandırıcısı modelidir. Sonuçlar, ML ve NLP yöntemlerinin yazılım kalite metriklerinin sınıflandırma sürecinde etkin bir şekilde kullanılabileceğini göstermektedir.

Anahtar Kelimeler: Makine öğrenmesi, Doğal dil işleme, ISO/IEC 25010, Smote, Çok sınıflı sınıflandırma

[a] *20239258007@cumhuriyet.edu.tr* *ORCID: 0009-0009-6271-036X* | [b] *hakankekul@cumhuriyet.edu.tr* *ORCID: 0000-0001-6269-8713*

How to Cite: Sönmez S, Kekül H (2024) Multi-Class Classification of ISO/IEC 25010 Software Quality Metrics Using User Feedback, Journal of Engineering Faculty, 2(2): 214-222

Introduction

Software quality is an important factor in all stages of the development process. Quality orientated approaches in requirements specification and design phases, early detection of errors and reduces costs [1]. Since improving the quality of software will directly affect both user satisfaction and the success of the business, software developers and companies need some methods to objectively measure the quality of their products[2]. ISO/IEC 25010 is an internationally recognized standard for assessing software quality. It consists of two main models: software product quality and quality of use [3]. Standardized, online health portals, mobile has been applied to evaluate various systems, including applications and e-commerce Web sites [4], [5], [6]. The standard includes eight metrics that describe different aspects of software: functional suitability, performance, compatibility, usability, reliability, security, maintainability and portability. Each of these features evaluates a specific aspect of the software [7]. The usability metric measures how easily a software can be learned and used by users, while the security metric examines how well the software defends against unauthorized access. The main reasons for the widespread use of these metrics are that they comprehensively evaluate many aspects of software and offer a different perspective. Research shows that sustainability, performance efficiency and usability are the most frequently evaluated quality metrics [8].

The use of user comments in the evaluation of software quality is becoming increasingly important. User experiences are becoming more important as they consist of real-time data. Research shows that analyzing user feedback can provide valuable insights into application quality and guide development [9]. The importance of user preferences in mobile application selection has increased; factors such as language, price, performance and user feedback have become important criteria. [10]. However, By revealing hidden elements in the comments associated with the ISO/IEC 25010 standard, valuable information about various aspects of the software, such as functionality or security, can be obtained [11]. These findings suggest that user feedback plays a very important role in improving software quality. Automated classifiers can help classify user statements according to ISO/IEC 25010 quality metrics, reducing manual effort in requirements elicitation [12]. Therefore, natural language processing and machine learning techniques can be an effective tool for software quality assessment by playing an important role in analyzing user comments [13].

The following research questions are addressed in this article.

R.Q.1: Which machine learning algorithm provides the highest accuracy in classifying ISO/IEC 25010 software quality metrics based on user feedback?

R.Q.2: How does the SMOTE method applied to unbalanced data sets affect the performance of classifiers in the classification of ISO/IEC 25010 software quality metrics?

R.Q.3: How does the classification of quality metrics with user comments contribute to the software quality assessment process?

The paper is designed as follows: firstly, the importance of software quality and the role of the ISO/IEC 25010 standard in software quality assessment are discussed. Then, similar studies in the literature are summarized and in the methodology section, the processing of the dataset, imbalance removal with SMOTE and the machine learning algorithms used are discussed in detail. In the findings section, the classification performance of each algorithm is compared. In the Discussion section, the contribution of the findings to software quality assessment processes is evaluated. In the last part, the conclusions obtained from the study and future studies are emphasized.

Literature Review

Haoues et al. [14] In their study, they classified mHealth app reviews according to ISO/IEC 25010 features and sensitivity polarity and achieved a high accuracy rate. Similarly Zahra and Kraugusteeliana [15] In their study, they analyzed the performance of a digital banking application by defining security as a critical feature and using ISO/IEC 25010. Şenkal et al. [16] They analyzed the DevOps pipeline from the perspective of ISO/IEC 25010 and identified gaps in the quality framework. Yetiş and Das [17] focused on software product metrics, describing source code and class-based metrics and demonstrating their implementation through a Java-based library.

Tuna [18] In his study, he investigated the classification of emotions in mobile app customer feedback using machine learning methods and emphasized the potential of voluntary online feedback as a tool for accurately understanding customers. Ramadhan and Hartomo [19] In their study, they evaluated the quality of a disaster information system website using the WebQual 4.0 method, which assesses usability, information quality, interaction quality and user satisfaction. Their findings emphasize the importance of improving information quality to increase user satisfaction. Onaran and Gençtürk [20] When they evaluated the service quality of mobile municipality applications in the context of e-government, they found that while ease of use was appreciated, reliability and other dimensions were perceived negatively.

Yalçın and Yağlı [21] In their study, they developed a hierarchical quality model based on ISO/IEC 25010 quality model and evaluated the website quality of technology stores with this model. The study proposes a new quality assessment model for website evaluation and provides guidance for technology stores to design better quality websites for their users. Keibach and Shayesteh [22] In their study, they examined the capabilities and limitations of software tools for climate adaptation in landscape design. This assessment, based on the ISO/IEC 25010 framework, focussed on quality attributes of software such as functionality, reliability, performance efficiency, usability, compatibility and information quality.

Majumdar et al. [23] developed a framework for classifying code comments as useful, partially useful or not useful for software maintenance. Yahya et al. [24] proposed a hybrid deep learning model to detect and classify non-functional requirements in mobile application reviews. Khan et al. [25] By analysing user feedback from low-scoring applications to identify common problems, they compared various ML and deep learning (DL) algorithms to classify these problems. Botchway et al. [26] They used the Analytic Hierarchy Process to evaluate software quality attributes and found that the most important ones are maintainability, security and testability.

These studies demonstrate the value of user feedback in improving software quality and highlight the effectiveness of ML techniques in analysing user comments for software quality assessment.

Methodology

This section describes the methodology used in the study. Within the scope of the study, a series of data processing steps and various machine learning algorithms were applied to classify user feedback according to ISO/IEC 25010 software quality metrics. After the data were first cleaned and normalised, the class imbalance was removed using the SMOTE method and then the results were evaluated using five different classifiers. Model evaluations were made using performance metrics such as accuracy, precision, recall and f1-score. Figure 1 shows the flowchart.

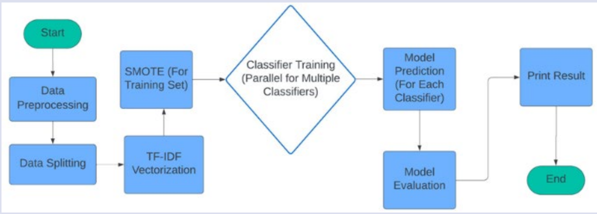

Figure 1. Process flow chart

Data Set

Haoues et al. [14] The data set created by, consists of user feedbacks, emotions and quality metrics columns. Feedbacks are categorized into three categories: positive, negative and neutral. The dataset contains feedbacks of 1681 users. In addition, these feedbacks were associated with quality metrics within the framework of the ISO/IEC 25010 software quality model. An example of the data set is shown in Table 1. In addition, Figure 2 shows the class distribution of quality metrics.

Table 1. An example from the data set

Body	Sentiment	Quality
The women's health portion isn't ac…	negative	Functional Suitability
I loved this app because it worked with other …	positive	Compatibility
Trying to connect after buying an galaxy ac…	negative	Performance
Really good app and easy to use.	positive	Usability
Can't get past the login page - just get an…	negative	Security

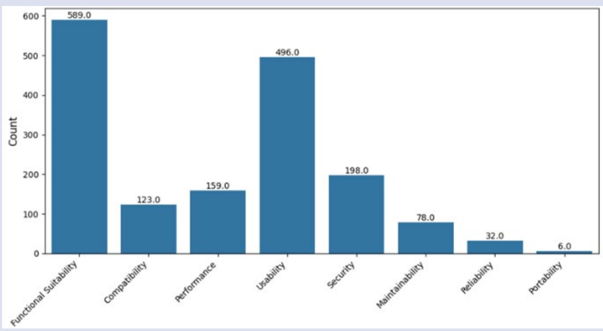

Figure 2. Class distribution of quality metrics

Processing of Comments

The feedbacks to be used for analysis should be subjected to a series of pre-processing. In this study, the following were performed as data cleaning and organisation steps.

- Unicode normalisation check was performed.
- URLs in the texts were detected and removed with regular expressions regex.
- Emojis have been removed.
- All texts have been converted to lower case.
- The words in the texts have been reduced to roots.
- Stop words have been removed.
- Numeric characters and punctuation marks have been removed and only alphabetic characters have been retained.

Synthetic Data Generation

SMOTE is a popular method to address class imbalance in ML [27]. Generates synthetic samples for the minority class by linear interpolation between existing samples and their nearest neighbours [28]. When the class distributions of the data are analysed, it is observed that there is a significant class imbalance in the "Quality" categories. This imbalance may cause classes to be underrepresented by the model and especially minority classes to have difficulty in the learning process. Especially "Functional Suitability" (589) and "Usability" (496) classes have large sample groups, while some classes such as "Portability" (6) and "Reliability" (32) are represented by very few samples. This may lead to the model not being able to learn small classes sufficiently and the prediction performance for these classes may decrease. [29].

In order to reduce the negative effects of such class imbalances and to enable the model to learn small classes more effectively, the SMOTE method was applied. [30]. In the "Quality" category, a balanced distribution was achieved with 460 samples for each class. In addition to the classes such as "Functional Suitability" and "Usability" which initially had large sample groups, small classes such as "Portability" and "Reliability" were also equalized with 460 samples. This balance ensured that all quality attributes were learnt by the model with equal weight and eliminated the under-representation problems of small classes.

Classification Models

In this study, five different classification algorithms were used to classify user comments according to ISO/IEC 25010 software quality metrics.

Extra Trees Classifier (ETC)

ETC is a method that combines predictions by combining multiple decision trees [31]. This approach reduces overfitting while improving generalization [32]. The algorithm does not spend much time finding the best split, which makes it faster [33]. Random selection of splits is a feature that speeds up the training process [34]. This classification algorithm aims to improve prediction accuracy and control overfitting by applying a series of randomized decision tree techniques to subsamples of the dataset [35].

Gaussian Process Classifier (GPC)

GPC is a supervised ML method for solving regression and probabilistic classification problems [36]. Supports multi-class classification [37]. It does this by training and predicting each class against the others. Uses the one class versus all other classes approach when performing the classification process [38].

MLP Classifier (MLPC)

This model is known as a multilayer perceptron classifier and uses the stochastic gradient descent method to optimize the log-loss function [39]. MLPC is an artificial neural network model and supports multi-class classification.

Bernoulli Naive Bayes Classifier (BNBC)

BNBC supports multi-class classification. It performs probability calculations for each class and predicts the class with the highest probability. Particularly suitable for datasets containing binary (0/1) features, but can also work with multi-class datasets [40].

Support Vector Classifier (SVC)

SVC is a support vector machines (SVM) algorithm and can work with multi-class datasets. Builds and predicts separate models for each class using the "one-vs-one" strategy for multi-class datasets [41].

Results

In this study, several machine learning algorithms are applied for multi-class classification of ISO/IEC 25010 software quality metrics using user feedback. Metrics such as precision, recall and f1-score were used to evaluate the classification performances and the weighted average results for each algorithm were compared.

Pre-Smote Classification Results

In this section, the classification results obtained with the dataset consisting of raw data before data balancing with SMOTE are presented in detail. Table 2 presents the classification results of the different classification algorithms used.

Table 2. Weighted average comparison table calculated before smote

Algoritm	Precision	Recall	F1 Score	Accuracy
GPC	0.80	0.73	0.70	0.7329
MLPC	0.79	0.80	0.79	0.7982

ETC	0.87	0.86	0.85	0.8575
BNBC	0.61	0.58	0.52	0.5786
SVC	0.85	0.80	0.78	0.7982

A comparative evaluation of the five algorithms used is presented below:

GPC: This algorithm performed poorly compared to the other models with 73% recall and 70% f1-score. Although the precision is 80%, the relatively low recall value indicates imbalances in some classes.

MLPC: It gave very balanced results with 79% precision, 80% recall and 79% f1-score. This model can be preferred because it provides a balanced distribution between classes and shows successful results in terms of overall performance.

ETC: It is the highest performing model with 87% precision, 86% recall and 85% f1-score. Especially considering the precision and recall values, this model provided the best result in terms of both accuracy and overall classification success.

BNBC: This algorithm showed the lowest performance compared to the others (61% precision, 58% recall, 52% f1-score). The lack of sufficient data in some classes and the poor discrimination power of the model between classes led to low results.

SVC: provided a balanced result with 85% precision, 80% recall and 78% f1-score. This model, which has a particularly high precision value, gave satisfactory results in terms of overall accuracy, but imbalances were observed in some classes. According to the results, the ETC algorithm showed the highest success with an accuracy of 85.75%. This model provided the best performance despite the data imbalance. It is followed by MLPC and SVC with an accuracy of 79.82%. Although these two models provide a reasonable balance between the classes, they resulted in less accuracy. The GPC algorithm performed slightly lower with an accuracy of 73.29%. Data imbalance may be one of the factors affecting the performance of this model. The model with the lowest accuracy was BNBC with 57.86%. This model may not be a suitable choice for this type of multi-class dataset, as it performs poorly especially in data imbalances.

Considering all the results, the ETC algorithm showed the highest performance in the classification of ISO/IEC 25010 quality metrics based on user feedback. This model has achieved the most successful results in extracting quality metrics from user feedback, especially due to its balanced precision and recall ratios. Among the other models, MLPC and SVC also gave remarkable results, but the low recall rates for some classes reveal the limitations of these models. As a result, the ETC algorithm stands out as the most successful model.

Post-Smote Classification Results

In this section, the success rates of the classification models obtained with the data generated by eliminating the imbalance of the data set are presented. It is an expected result that the classification success increases as a result of the balanced data set in multi-class classification. The obtained classification results are given in Table 3.

Table 3. Weighted average comparison table calculated after smote

Algoritm	Precision	Recall	F1 Score	Accuracy
GPC	0.87	0.86	0.86	0.8605
MLPC	0.81	0.81	0.81	0.8100
ETC	0.87	0.87	0.87	0.8694
BNBC	0.74	0.72	0.72	0.7210
SVC	0.85	0.80	0.79	0.8011

The GPC and BNBC models showed the largest increase in accuracy after SMOTE. This shows that these models were significantly affected by the data imbalance and SMOTE application eliminated this effect. ETC slightly improves its performance after SMOTE and stands out as the most stable model with an accuracy of 87%. Other models (MLPC and SVC) showed a more limited improvement. Figure 3,4,5,6,7 shows the confusion matrix plots of the classifiers.

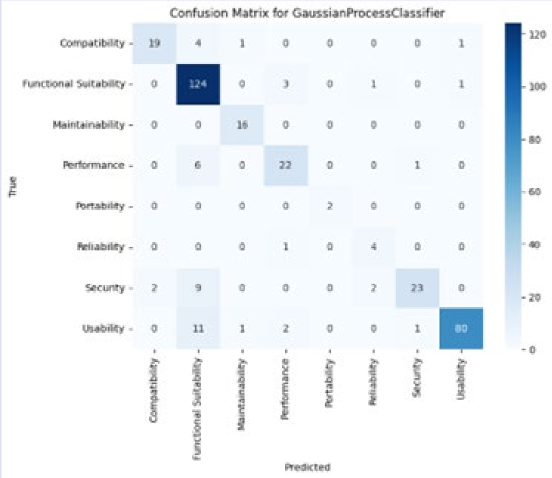

Figure 3. Confusion matrix of the GPC

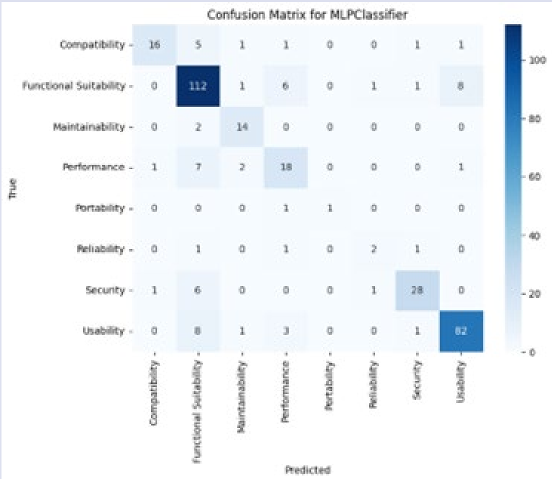

Figure 4. Confusion matrix of the MLPC

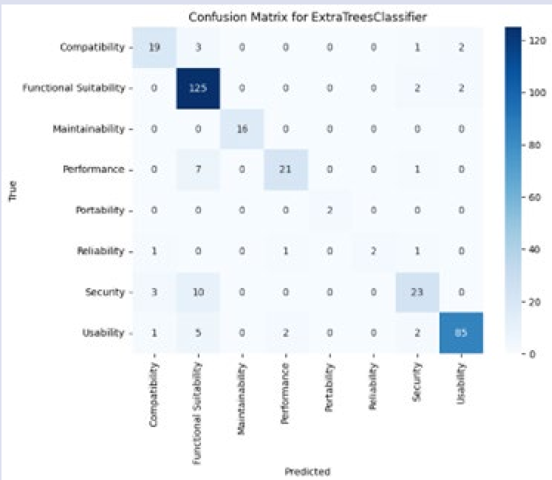

Figure 5. Confusion matrix of the ETC

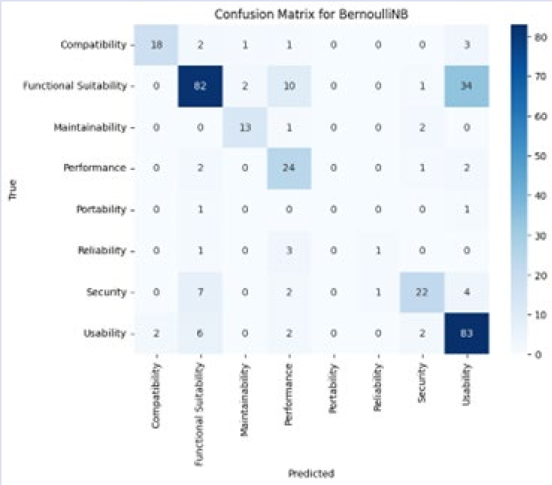

Figure 6. Confusion matrix of the BNBC

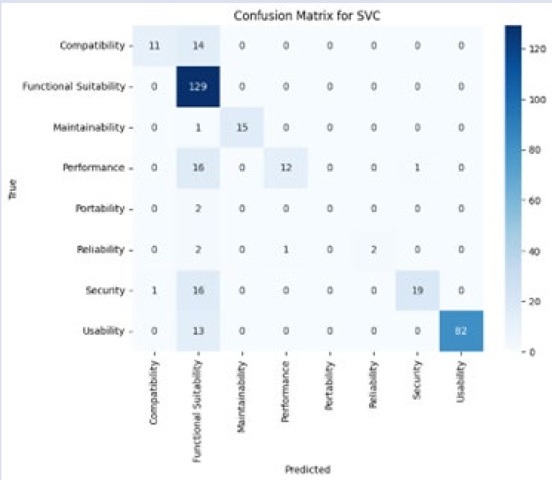

Figure 7. Confusion matrix of the SVC

Discussion

This study can be an important guide for future research by contributing to the determination of the most appropriate algorithm for software quality classification problems. The findings obtained within the scope of the study reveal that user comments can be successfully classified within the framework of the ISO/IEC 25010 software quality model and that this classification is useful. The high accuracy rates of algorithms such as ETC and GPC show that quality metrics can be successfully extracted from user comments. These findings reinforce the relationship between ISO/IEC 25010 standards and user feedback and allow software quality to be evaluated based on user experiences. However, it has been observed that some metrics are more difficult to extract from user feedback, creating a limitation that can be addressed with further research.

The performance of algorithms varies depending on the dataset used and the characteristics of the algorithm. For example, the low accuracy rate of the BNBC algorithm emphasizes that more advanced algorithms should be preferred in such text classification problems. In the future, it may be useful to extend the method and use different algorithms to work with more complex data sets and metrics.

Conclusion

In this study, five different machine learning algorithms were used to classify feedback from users according to ISO/IEC 25010 software quality metrics. GPC, MLPC, ETC, BNBC and SVC algorithms have been tested. According to the results, after applying the smote, ETC provided the highest accuracy rate of 87%. While GPC offered a close accuracy rate of 86.05%, the BNBC algorithm showed the lowest performance with 72.10%. These findings show that ETC and GPC algorithms, in particular, have a strong potential for making meaningful inferences about quality metrics from user comments. On the other hand, MLPC, which gives lower accuracy rates of 81%, and SVC algorithms, which give lower accuracy rates of 80.11%, were also evaluated, but did not show superior performance. As a result, this study emphasized the effect of different algorithms on software quality classification and revealed that ETC is a model that can be recommended for such classification problems. Future studies can investigate ways to improve the performance of these algorithms by testing them on larger data sets. Additionally, it appears that some ISO/IEC 25010 metrics are quite difficult to extract based on user comments. Therefore, the relationships between user comments and metrics should be further investigated. More studies should be conducted on extracting valuable comments from user comments and different dimensions of evaluation in comments.

Acknowledgements

This study was produced as a part of the Master's thesis titled "Classification of ISO/IEC 25010 Software Quality Metrics Using User Comments" prepared by Sedat SÖNMEZ.

References

[1] D. YILDIZ, 'Bilgi Sistemi Yazılım Geliştirme Yaşam Döngüsü Safhalarından Gereksinim Belirleme ve Sistem Tasarımında Kalite Odaklılık: Üç Proje İncelemesi', Bilişim Teknol. Derg., vol. 15, no. 1, pp. 55–64, 2022, doi: 10.17671/gazibtd.871411.

[2] J. P. Miguel, D. Mauricio, and G. Rodríguez, 'A Review of Software Quality Models for the Evaluation of Software Products', Int. J. Softw. Eng. Appl., vol. 5, no. 6, pp. 31–53, 2014, doi: 10.5121/ijsea.2014.5603.

[3] M. D. Mulyawan, I. N. S. Kumara, I. B. A. Swamardika, and K. O. Saputra, 'Kualitas Sistem Informasi Berdasarkan ISO/IEC 25010: Literature Review', Maj. Ilm. Teknol. Elektro, vol. 20, no. 1, p. 15, Mar. 2021, doi: 10.24843/mite.2021.v20i01.p02.

[4] A. Hussain and E. O. C. Mkpojiogu, 'An application of the ISO/IEC 25010 standard in the quality-in-use assessment of an online health awareness system', J. Teknol., vol. 77, no. 5, pp. 9–13, 2015, doi: 10.11113/jt.v77.6107.

[5] R. D. Jayanto, 'EVALUASI KUALITAS APLIKASI MOBILE KAMUS ISTILAH JARINGAN PADA PLATFORM ANDROID DENGAN STANDAR ISO/IEC 25010', Elinvo (Electronics, Informatics, Vocat. Educ., vol. 2, no. 2, pp. 178–182, Dec. 2017, doi: 10.21831/elinvo.v2i2.17311.

[6] N. YALÇIN and B. ŞİMŞEK YAĞLI, 'TEKNOLOJİ MAĞAZALARININ ISO 25010 KALİTE MODELİNE DAYALI WEBSİTESİ KALİTE DEĞERLENDİRMESİNİN ÇOK KRİTERLİ ANALİZİ: TÜRKİYE ÖRNEĞİ', Uluslararası İktisadi ve İdari İncelemeler Derg., pp. 57–76, Feb. 2020, doi: 10.18092/ulikidince.557263.

[7] K. Moumane, A. Idri, F. El Aouni, J. Laghnimi, N. C. Benabdellah, and O. Hamal, 'ISO/IEC 25010-based Quality Evaluation of Three Mobile Applications for Reproductive Health Services in Morocco', Clin. Exp. Obstet. Gynecol., vol. 51, no. 4, 2024, doi: 10.31083/j.ceog5104088.

[8] L. Carlos Belezia, M. Fatima Ludovico de Almeida, M. Falco, and G. Robiolo, 'Building a Catalogue of ISO/IEC 25010 Quality Measures Applied in an Industrial Context', J. Phys. Conf. Ser., vol. 1828, no. 1, p. 012077, Feb. 2021, doi: 10.1088/1742-6596/1828/1/012077.

[9] A. Di Sorbo, G. Grano, C. Aaron Visaggio, and S. Panichella, 'Investigating the criticality of user-reported issues through their relations with app rating', J. Softw. Evol. Process, vol. 33, no. 3, pp. 1–21, 2021, doi: 10.1002/smr.2316.

[10] T. EREN, 'Mobil Uygulama Seçiminde Etkili Olan Kriterlerin Belirlenmesi ve Örnek Uygulama', Istanbul J. Econ. / İstanbul İktisat Derg., vol. 70, no. 1, pp. 113–139, Jun. 2020, doi: 10.26650/istjecon2019-0022.

[11] I. Atoum and C. H. Bong, 'Measuring Software Quality in Use: State-of-the-Art and Research Challenges', vol. 17, pp. 4–15, 2015, [Online]. Available: http://arxiv.org/abs/1503.06934

[12] R. dos Santos, K. Villela, D. T. Avila, and L. H. Thom, 'A practical user feedback classifier for software quality characteristics', Proc. Int. Conf. Softw. Eng. Knowl. Eng. SEKE, vol. 2021-July, pp. 340–345, 2021, doi: 10.18293/SEKE2021-055.

[13] B. El Fhel and A. Idri, 'Quality In-Use of Mobile Geographic Information Systems for Data Collection', Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 11, pp. 903–913, 2023, doi: 10.14569/IJACSA.2023.0141192.

[14] M. Haoues, R. Mokni, and A. Sellami, 'Machine learning for mHealth apps quality evaluation: An approach based on user feedback analysis', Softw. Qual. J., vol. 31, no. 4, pp. 1179–1209, Dec. 2023, doi: 10.1007/s11219-023-09630-8.

[15] M. N. Zahra and K. Kraugusteeliana, 'Analisis Kualitas Performa Aplikasi Digital Banking X Menggunakan Framework ISO 25010', J. Teknol. Inf. dan Ilmu Komput., vol. 10, no. 3, pp. 483–490, Jul. 2023, doi: 10.25126/JTIIK.20231036326.

[16] 'Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi » Makale » DevOps İşletim Hattının Kalite Bakış Açısı ile İrdelenmesi ve ISO/IEC 25010 Boşluk Analizinin Gerçekleştirilmesi'. Accessed: Oct. 20, 2024. [Online]. Available: https://dergipark.org.tr/tr/pub/tbbmd/issue/78526/1118200

[17] A. KAHVECİ YETİŞ and R. DAŞ, 'Yazılım Ürün Ölçütlerinin Uygulamalı İncelenmesi', Fırat Üniversitesi Mühendislik Bilim. Derg., vol. 34, no. 2, pp. 635–645, 2022, doi: 10.35234/fumbd.1114056.

[18] M. F. TUNA, 'MOBİL UYGULAMA MÜŞTERİ GERİ BİLDİRİMİNDEKİ DUYGULARIN MAKİNE ÖĞRENMESİ YÖNTEMLERİYLE SINIFLANDIRILMASI', J. Bus. Commun. Stud., vol. 1, no. 1, pp. 83–103, 2022, doi: 10.29228/JOBACS.63080.

[19] M. R. Ramadhan and K. D. Hartomo, 'Evaluasi Kualitas Website Menggunakan Webqual 4.0 (Studi Kasus: Sistem Informasi Kebencanaan Kabupaten Boyolali)', J. Transform., vol. 19, no. 2, p. 138, Jan. 2022, doi: 10.26623/TRANSFORMATIKA.V19I2.4195.

[20] B. ONARAN and K. K. GENÇTÜRK, 'Mobil Belediye Aplikasyonlarında Hizmet Kalitesi: İzmir Örneği', Uluslararası Yönetim Akad. Derg., vol. 5, no. 1, pp. 16–31, 2022, doi: 10.33712/mana.1039540.

[21] N. YALÇIN and B. ŞİMŞEK YAĞLI, 'TEKNOLOJİ MAĞAZALARININ ISO 25010 KALİTE MODELİNE DAYALI WEBSİTESİ KALİTE DEĞERLENDİRMESİNİN ÇOK KRİTERLİ ANALİZİ: TÜRKİYE ÖRNEĞİ', Uluslararası İktisadi ve İdari İncelemeler Derg., pp. 57–76, Feb. 2020, doi: 10.18092/ulikidince.557263.

[22] E. Keibach and H. Shayesteh, 'BIM for Landscape Design Improving Climate Adaptation Planning: The Evaluation of Software Tools Based on the ISO 25010 Standard', Appl. Sci., vol. 12, no. 2, Jan. 2022, doi: 10.3390/APP12020739.

[23] S. Majumdar, A. Bansal, P. P. Das, P. D. Clough, K. Datta, and S. K. Ghosh, 'Automated evaluation of comments to aid software maintenance', J. Softw. Evol. Process, vol. 34, no. 7, Jul. 2022, doi: 10.1002/SMR.2463.

[24] A. E. Yahya, A. Gharbi, W. M. S. Yafooz, and A. Al-Dhaqm, 'A Novel Hybrid Deep Learning Model for Detecting and Classifying Non-Functional Requirements of Mobile Apps Issues', Electronics, vol. 12, no. 5, Mar. 2023, doi: 10.3390/ELECTRONICS12051258.

[25] N. D. Khan, J. A. Khan, J. Li, T. Ullah, and Q. Zhao, 'Mining software insights: uncovering the frequently occurring

issues in low-rating software applications', PeerJ Comput. Sci., vol. 10, p. e2115, Jul. 2024, doi: 10.7717/PEERJ-CS.2115/SUPP-1.

[26] B. I. Belinda, A. A. Emmanuel, N. Solomon, and A. B. Kayode, 'Evaluating Software Quality Attributes using Analytic Hierarchy Process (AHP)', Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 3, pp. 165–173, 2021, doi: 10.14569/IJACSA.2021.0120321.

[27] G. A. Pradipta, R. Wardoyo, A. Musdholifah, I. N. H. Sanjaya, and M. Ismail, 'SMOTE for Handling Imbalanced Data Problem : A Review', in 2021 6th International Conference on Informatics and Computing, ICIC 2021, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICIC54025.2021.9632912.

[28] P. Soltanzadeh and M. Hashemzadeh, 'RCSMOTE: Range-Controlled synthetic minority over-sampling technique for handling the class imbalance problem', Inf. Sci. (Ny)., vol. 542, pp. 92–111, Jan. 2021, doi: 10.1016/j.ins.2020.07.014.

[29] E. Sugawara and H. Nikaido, 'Properties of AdeABC and AdeIJK efflux systems of Acinetobacter baumannii compared with those of the AcrAB-TolC system of Escherichia coli', Antimicrob. Agents Chemother., vol. 58, no. 12, pp. 7250–7257, 2014, doi: 10.1128/AAC.03728-14.

[30] A. A. Arifiyanti, A. A. Arifiyanti, and E. D. Wahyuni, 'SMOTE: METODE PENYEIMBANG KELAS PADA KLASIFIKASI DATA MINING', Scan J. Teknol. Inf. dan Komun., vol. 15, no. 1, pp. 34–39, Feb. 2020, doi: 10.33005/scan.v15i1.1850.

[31] P. A. Manilal and A. C. Suthar, 'Object Based Image Classification of Satellite Images for Urban Areas', 2023. Accessed: Oct. 17, 2024. [Online]. Available: http://gtusitecirculars.s3.amazonaws.com/uploads/Final Synopsis_465548.pdf

[32] A. Pagliaro, 'Forecasting Significant Stock Market Price Changes Using Machine Learning: Extra Trees Classifier Leads', Electron., vol. 12, no. 21, p. 4551, Nov. 2023, doi: 10.3390/electronics12214551.

[33] D. Baby, S. J. Devaraj, J. Hemanth, and M. M. Anishin Raj, 'Leukocyte classification based on feature selection using extra trees classifier: A transfer learning approach', Turkish J. Electr. Eng. Comput. Sci., vol. 29, pp. 2742–2757, 2021, doi: 10.3906/elk-2104-183.

[34] N. Bhandari, 'How does ExtraTreesClassifier reduce the risk of overfitting?' [Online]. Available: https://medium.com/@namanbhandari/extratreesclassifier-8e7fc0502c7

[35] 'Extra Trees Classifier', scikit-learn Machine Learning in Python. [Online]. Available: https://scikit-learn.org/dev/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[36] O. O. Petinrin, X. Li, and K. C. Wong, 'Particle Swarm Optimized Gaussian Process Classifier for Treatment Discontinuation Prediction in Multicohort Metastatic Castration-Resistant Prostate Cancer Patients', IEEE J. Biomed. Heal. Informatics, vol. 26, no. 3, pp. 1309–1317, 2022, doi: 10.1109/JBHI.2021.3103989.

[37] I. Achituve, A. Navon, Y. Yemini, G. Chechik, and E. Fetaya, 'GP-Tree: A Gaussian Process Classifier for Few-Shot Incremental Learning', Proc. Mach. Learn. Res., vol. 139, pp. 54–65, Feb. 2021, Accessed: Oct. 17, 2024. [Online]. Available: https://arxiv.org/abs/2102.07868v4

[38] 'Gaussian Process Classification'. [Online]. Available: https://scikit-learn.org/1.5/modules/gaussian_process.html#gaussian-processes

[39] 'MLP Classifier'. [Online]. Available: https://scikit-learn.org/dev/modules/generated/sklearn.neural_network.MLPClassifier.html

[40] O. Iparraguirre-Villanueva, K. Espinola-Linares, R. O. Flores Castañeda, and M. Cabanillas-Carbonell, 'Application of Machine Learning Models for Early Detection and Accurate Classification of Type 2 Diabetes', Diagnostics, vol. 13, no. 14, 2023, doi: 10.3390/diagnostics13142383.

[41] M. M. Nishat, F. Faisal, T. Hasan, M. F. Bin Karim, Z. Islam, and M. R. K. Shagor, 'An Investigative Approach to Employ Support Vector Classifier as a Potential Detector of Brain Cancer from MRI Dataset', Proc. Int. Conf. Electron. Commun. Inf. Technol. ICECIT 2021, pp. 1–4, 2021, doi: 10.1109/ICECIT54077.2021.9641168.